ENVIRONMENT MODELING FROM IMAGES TAKEN BY A LOW COST CAMERA

Maxime Lhuillier

LASMEA, UMR 6602 Université Blaise Pascal/CNRS, 63177 Aubière Cedex, France, http://maxime.lhuillier.online.fr

Commission III/1

KEY WORDS: catadioptric camera, multi-view geometry estimation, environment reconstruction, vision system

ABSTRACT:

This paper describes a system to generate 3D model from image sequence taken in complex environment including variable ground surface, buildings and trajectory loops. Here we use a $1000 catadioptric camera and approximate knowledge of its calibration. This contrasts to current systems which rely on more costly hardware such as the (calibrated) spherical vision camera Ladybug. All steps of the method are summarized. Experiments include a campus reconstruction from thousands of images.

1 INTRODUCTION

The automatic 3D modeling of environments from image sequence is a long-term and still active field of research. Wide view field camera is a natural choice for ground-based sequence. Current systems include multi-camera and accurate GPS and INS (Pollefeys et al., 2008) at high cost (>$100K), the Ladybug multi-camera (www.ptgrey.com, 2010) at medium cost (≈$12K). Here we use a catadioptric camera at low cost (≈$1K): a mirror of revolution (www.0 −360.com, 2010) mounted on a perspective still camera (Nikon Coolpix 8700) thanks to adapter ring. The medium and high cost systems are more convenient since they provide video sequences and wide view field without sacrificing image resolution.

The first step is the estimation of successive camera poses using Structure-from-Motion (SfM). Recent work (Micusik and Kosecka, 2009) suggests that bundle adjustment (BA) is not needed if several good experimental conditions are met: large resolution, wide view field, and accurate knowledge of calibration. Here we do not require accurate calibration since this depends on both mirror profile (mirror manufacturer may not like to reveal this) and the pose between the perspective camera and the mirror. Furthermore, we would like to avoid calibration pattern handling for end-users. For these reasons, BA is needed to estimate simultaneously camera poses, reconstructed points and intrinsic parameters.

Drift or error accumulation occurs in SfM of long image sequences. It should be detected between images taken at similar locations in the scene (Anan and Hartley, 2005, Havlena et al., 2009) and removed using BA (Cornelis et al., 2004). Here we remove drift using constrained bundle adjustment (CBA) based on (Triggs et al., 2000), instead of a re-weighted version of the standard BA (Cornelis et al., 2004) which relies considerably on heuristic initialization.

The next step is the estimation of the 3D scene. Like (Pollefeys et al., 2008, Micusik and Kosecka, 2009), we apply dense stereo method on a small number of M consecutive images, iterate this process several times along the sequence, and merge the obtained view-centered 3D models into the global and final 3D model. Furthermore, we use an over-segmentation in the reference image of view-centered model in conjunction with dense stereo (Zitnick and Kang, 2007, Micusik and Kosecka, 2009). Super-pixels (small regions) are useful to reduce stereo ambiguity, to constrain depth discontinuities at super-pixel borders selected among image contours, to reduce computational complexity.

Our over-segmentation is defined by triangle mesh in image such that (1) super-pixel is a list of connected triangles and (2) triangles of the view-centered model are back-projected triangles of the image mesh. In work (Zitnick and Kang, 2007, Micusik and Kosecka, 2009), super-pixel is pixel list and is unused by surface meshing. Our choice has several advantages. Mesh regularizes super-pixel shape and defines the resolution of final reconstruction by view field sampling. This is useful to compress large scene. Besides, we obtain triangles in 3D such that the consistency with image contours can not be degraded by depth error unlike (Chai et al., 2004). This is not a luxury because depth estimation is difficult in uncontrolled environment. Our super-pixels are not restricted to be planar in 3D contrary to those in (Zitnick and Kang, 2007, Micusik and Kosecka, 2009).

The last step is filtering of triangles in view-centered models. It reduces the redundancy and removes the most inaccurate and unexpected triangles. Here we accept non-incremental method with complexity greater than linear in the number of camera poses, since the main calculations are done in the previous step (which has linear complexity and is parallelizable using multi-cores).

This paper improves work (Lhuillier, 2008a, Lhuillier, 2008b) thanks to polygons for super-pixels, drift removal using CBA, accelerations for larger sequence (feature selection in the SfM step, complexity handling in the triangle filtering step), redundancy reduction, experiments on more challenging sequence.

2 OVERVIEW OF THE RECONSTRUCTION METHOD

This Section has six parts describing camera model, Structure-from-Motion, drift removal, over-segmentation mesh, view-cen--tered model and triangle filtering.

2.1 Camera Model

A single view-point camera model with a general radial distortion function and a symmetry axis is used (Lhuillier, 2008a). It simplifies the reconstruction process for non-single view-point camera (if any), assuming that depth is large enough.

We assume that the projection of the whole view field is delimited by two concentric circles which can be detected in images. Furthermore, the mirror manufacturer provides the lower and upper bounds of the “ray angle” between observation ray and the symmetry axis. The initial calibration is that of equiangular camera: the mapping from the ray angle of 3D point to the distance between the point projection and the circle center is linear.

2.2 Structure-from-Motion

Structure-from-Motion (Lhuillier, 2008a) is applied to estimate geometry (camera poses and a sparse cloud of 3D points) using the calibration initialization of Section 2.1: estimate the 2-view and 3-view geometries of consecutive images from matched Harris points (step 1) and then estimate the whole sequence geometry using bundle adjustment (BA) applied in a hierarchical framework (step 2). Another BA is applied to refine simultaneously radial distortion parameters and the 3D assuming that the radial distortion is constant in the whole sequence (step 3).

Although it is important to get a maximum number of reconstructed features for 3D scene modeling, we noticed that there are many more 3D points than needed to initialize the geometry in our wide view field context. Indeed, this is not uncommon to have more than 2000 features per outdoor image involved in BA, and this implies a waste of computation time. So the number n_f of features per image is limited to 500 in all BAs of steps 1-2-3: 3D points are randomly selected and removed while n_f is larger than 500 in all images. In practice, this simple scheme holds a good point distribution in the view field. The 4^{th} step is the following: step 3 is applied a second time without n_f limit to get a maximum number of reconstructed features consistent with the poses and calibration.

Our BA is the sparse Levenberg-Marquardt method assuming that there are more structure parameters than camera ones: it includes profile Choleski decomposition (Triggs et al., 2000) of the reduced camera system.

2.3 Drift Removal

Drift or error accumulation is unavoidable in the geometry estimation of long sequence. Methods (Havlena et al., 2009, Anan and Hartley, 2005, Cornelis et al., 2004) detect the drift between two reconstructed images i and j if these images are taken at similar locations. These methods also provide list $L_{i,j}$ of point matches between i and j, which is used to remove drift. Without drift removal, scene part visible in i and j is reconstructed twice.

Adequate BA and its initialization are applied to remove reconstruction duplicates while maintaining low re-projection errors in the whole sequence of images $\{0, 1 \cdots n-1\}$. Once the 2D feature match list $L_{i,j}$ is given for pair $\{i,j\}$, we remove the drift between i and j as follows. First, we choose integer k such that the neighborhood $\mathcal{N}(i)$ of i is the list $\{i-k \cdots i \cdots i+k\}$. Second, $\mathcal{N}(i)$ and its data (3D geometry and image features) are duplicated in images $\mathcal{N}(n+k) = \{n \cdots n+2k\}$ such that images $n+k$ and i are the same. Third, we use RANSAC to fit the similarity transformation s of 3D points matched by $L_{i,j}$ and apply s to obtain $\mathcal{N}(n+k)$ geometry in the same basis as $\{0 \cdots n-1\}$ geometry. Fourth, $\{0 \cdots n+2k\}$ geometry is refined by BA taking into account $L_{n+k,j}$ ($L_{n+k,j}$ is a copy of $L_{i,j}$ with image index changes). Now $\mathcal{N}(n+k)$ geometry is the drift correction of $\mathcal{N}(i)$ geometry. Fifth, constrained bundle adjustment (CBA) is applied to minimize the global re-projection error subject to constraint $c(\mathbf{x}) = 0$, where $\mathbf{x}$ concatenates 3D parameters of $\{0 \cdots n+2k\}$ and $c(\mathbf{x})$ concatenates the drifts between poses of $\mathcal{N}(i)$ and $\mathcal{N}(n+k)$ (more details in Appendix B). At this point, the drift between i and j is removed but $\mathcal{N}(n+k)$ is redundant. Last, we remove data involving $\mathcal{N}(n+k)$ and apply BA to $\{0 \cdots n-1\}$ geometry by taking into account $L_{i,j}$.

This scheme is applied using a limit of $n_f = 500$ features per image to avoid waste of time as in Section 2.2, with the only difference that $L_{i,j}$ and $L_{n+k,j}$ are not counted by this limit.

2.4 Over-Segmentation Mesh

This mesh has the following purposes. It makes image-based simplification of the scene such that the view field is uniformly sampled. This is useful for time and space complexities of further processing and more adequate than storing depth maps for all images. Furthermore, it segments the image into polygons such that depth discontinuities are constrained to be at polygon borders. These borders are selected among image contours. If contours are lacking, borders are preferred on concentric circles or radial segments of the donut image. This roughly corresponds to horizontal and vertical depth discontinuities for standard orientation of the catadioptric camera (if its symmetry axis is vertical).

In short, the image mesh is build in four steps: initialization checkerboard (rows are concentric rings, columns have radial directions, cells are two Delaunay triangles), gradient edge integration (perturb vertices to approximate the most prominent image contours), optimization (perturb all vertices to minimize the sum, for all triangles, of color variances, plus the sum, for all vertices, of squared moduluses of umbrella operators), and polygon segmentation (triangles are regrouped in small and convex polygons). In practice, lots of polygons are quadrilaterals similar to those of the initialization checkerboard.

2.5 View-Centered 3D Model

View-centered 3D model is build from image mesh (Section 2.4) assuming that the geometry is known (Sections 2.1, 2.2 and 2.3).

Depth Map in the Reference Image We reproject catadioptric image onto the 6 faces of a virtual cube and apply match propagation (Lhuillier and Quan, 2002) to two parallel faces of two cubes. The depth map in the i^{th} image is obtained by chaining matches between consecutive images of $\mathcal{N}(i)$. In the next steps, the over-segmentation mesh in the i^{th} image is back-projected to approximate the depth map.

Mesh Initialization For all polygons in image i, a plane in 3D (or nil if failure) is estimated by a RANSAC procedure applied on depths available inside the polygon. A depth is inlier of tentative plane if the corresponding 3D point is in this plane up to thresholding (Appendix A). The best plane π defines 3D points which are the intersections between π and the observation rays of the polygon vertices in the i^{th} image. These 3D points are called “3D vertices of polygon” although the polygon is 2D.

For all edges e in image i, we define boolean b_e which will determine the connection of triangles in both edge sides. Since depth discontinuity is prohibited inside polygons, we initialize $b_e = 1$ if both triangles are in the same polygon (other cases: $b_e = 0$).

Connection Connections between polygons are needed to obtain a more realistic 3D model. Thus edge booleans are forced to 1 if neighboring polygons satisfy coplanarity constraint. For all polygons p with a plane in 3D, we collect in list L_p the polygons q in p-neighborhood (including p) such that all 3D vertices of q are in the plane of p up to thresholding (Appendix A). If the sum of solid angles of polygons in L_p is greater than a threshold, we have confidence in coplanarity between all polygons in L_p and we set $b_e = 1$ for all edges e between two polygons of L_p.

Hole Filling We fill hole H if its neighborhood N is coplanar. Both H and N are polygon lists. The former is a connected component of polygons without plane in 3D and the latter contains polygons with plane in 3D. Neighborhood N is coplanar if there is a plane π (generated by random samples of vertices of N) such that all 3D vertices in N are in π up to thresholding (Appendix A). If N is coplanar, all polygons of H get plane π and we set $b_e = 1$ for all edges between two triangles of $H \cup N$.

View-Centered Mesh in 3D Now, 3D triangles are generated by back-projection of triangles in the image mesh using polygon planes and edge booleans. Triangle t inside a polygon p with plane in 3D is reconstructed as follows. Let C_v be the circularly-linked list of polygons which have vertex v of t. We obtain sub-list(s) of C_v by removing the C_v-links between consecutive polygons which share edges e such that $b_e = 0$. A C_v-link is also removed if one of its two polygons has not plane in 3D. Let S_v^p be the sub-list which contains p. The 3D triangle of t is defined by its 3 vertices: the 3D vertex reconstructed for v is the mean of 3D vertices of the polygons in S_v^p which correspond to v.

Refinement Here we provide a brief overview of the refinement, which is detailed in (Lhuillier, 2008b). The view-centered mesh (3D triangles with edge connections) is parametrized by the depths of its vertices and is optimized by minimizing a weighted sum of discrepancy and smoothing terms. The discrepancy term is the sum, for all pixels in a triangle with plane in 3D, of the squared distance between the plane and 3D point defined by pixel depth (Appendix A). The smoothing term is the sum, for all edges which are not at image contour, of the squared difference between normals of 3D triangles in both edge sides. This minimization is applied several times by alternating with mesh operations "Triangle Connection" and "Hole Filling" (Lhuillier, 2008b).

2.6 Triangle Filtering

For all i, the method in Section 2.5 provides a 3D model centered at image i using images $\mathcal{N}(i)$. Now, several filters are applied on the resulting list of triangles to remove the most inaccurate and unexpected triangles.

Notations We need additional notations in this Section. Here t is a 3D (not 2D) triangle of the i^{th} view-centered model. The angle between two vectors $\mathbf{u}$ and $\mathbf{v}$ is angle$(\mathbf{u}, \mathbf{v}) \in [0, \pi]$. Let $\mathbf{d}_i, \mathbf{c}_i$ be the camera symmetry direction and center at the i^{th} pose in world coordinates ($\mathbf{d}_i$ points toward the sky). Let $U_j(\mathbf{v})$ be the length of major axis of covariance matrix $\mathtt{C}_\mathbf{v}$ of $\mathbf{v} \in \mathbb{R}^3$ as if $\mathbf{v}$ is reconstructed by ray intersection from $\mathbf{v}$ projections in images $\mathcal{N}(j)$ using Levenberg-Marquardt.

Uncertainty Parts of the scene are reconstructed in several view-centered models with different accuracies. This is especially true in our wide view field context where a large part of the scene is visible in a single image. Thus, the final 3D model can not be defined by a simple union of the triangle lists of all view-centered models. A selection on the triangles should be done.

We reject t if the i^{th} model does not provide one of the best available uncertainties from all models: if all vertices $\mathbf{v}$ of t have ratio $U_i(\mathbf{v})/\min_j U_j(\mathbf{v})$ greater than threshold u_0.

Prior Knowledge Here we assume that the catadioptric camera is hand-held by a pedestrian walking on the ground such that (1) the camera symmetry axis is (roughly) vertical (2) the ground slope is moderated (3) the step length between consecutive images and the height between ground and camera center do not change too much. This knowledge is used to reject unexpected triangles which are not in a "neighborhood of the ground".

A step length estimate is $s = \text{median}_i ||\mathbf{c}_i - \mathbf{c}_{i+1}||$. We choose angles α_t, α_b between $\mathbf{d}_i$ and observation rays such that $0 < \alpha_t < \frac{\pi}{2} < \alpha_b < \pi$. Triangle t is rejected if it is below the ground: if it has vertex $\mathbf{v}$ such that angle$(\mathbf{d}_i, \mathbf{v} - \mathbf{c}_i) > \alpha_b$ and height $\frac{1}{s}\mathbf{d}_i^T(\mathbf{v} - \mathbf{c}_i)$ is less than a threshold. The sky rejection does not depend on scale s. We robustly estimate the mean m and standard deviation σ of height $\mathbf{d}_i^T(\mathbf{v} - \mathbf{c}_i)$ for all vertex $\mathbf{v}$ of the i^{th} model such that angle$(\mathbf{d}_i, \mathbf{v} - \mathbf{c}_i) < \alpha_t$. Triangle t is rejected if it has vertex $\mathbf{v}$ such that angle$(\mathbf{d}_i, \mathbf{v} - \mathbf{c}_i) < \alpha_t$ and $\frac{1}{\sigma}(\mathbf{d}_i^T(\mathbf{v} - \mathbf{c}_i) - m)$ is greater than a threshold.

Reliability 3D modeling application requires additional filtering to reject "unreliable" triangles that filters above miss. These triangles includes those which are in the neighborhood of the line supporting the $\mathbf{c}_j, j \in \mathcal{N}(i)$ (if any). Inspired by a two-view reliability method (Doubek and Svoboda, 2002), we reject t if it has vertex $\mathbf{v}$ such that $\max_{j,k \in \mathcal{N}(i)}$ angle$(\mathbf{v} - \mathbf{c}_j, \mathbf{v} - \mathbf{c}_k)$ is less than threshold α_0. This method is intuitive: t is rejected if ray directions $\mathbf{v} - \mathbf{c}_j, j \in \mathcal{N}(i)$ are parallel.

Redundancy Previous filters provide a redundant 3D model insofar as scene parts may be reconstructed by several mesh parts selected in several view-centered models. Redundancy increases with threshold u_0 of the uncertainty-based filter and the inverse of threshold α_0 of the reliability-based filter. Our final filter decreases redundancy as follows: 3D triangles at mesh borders are progressively rejected in the decreasing uncertainty order if they are redundant with other mesh parts. Triangle t is redundant if its neighborhood intersects triangle of the j^{th} view-centered model ($j \neq i$). The neighborhood of t is the truncated pyramid with base t and three edges. These edges are the main axes of the 90% uncertainty ellipsoids of the t vertices $\mathbf{v}$ defined by $\mathtt{C}_\mathbf{v}$.

Complexity Handling We apply the filters above in the increasing complexity order to deal with large number of triangles (tens of millions in our case). Filters based on prior knowledge and reliability are applied first. Thanks to $\mathbf{c}_j$ and reliability angle α_0, we estimate radius r_i and center $\mathbf{b}_i$ of a ball which encloses the selected part of the i^{th} view-centered model: $\mathbf{b}_i = \frac{1}{2}(\mathbf{c}_{i-1} + \mathbf{c}_{i+1})$ and $\tan(\alpha_0/2) = ||\mathbf{c}_{i+1} - \mathbf{c}_{i-1}||/(2r_i)$ if $\mathcal{N}(i) = \{i-1, i, i+1\}$. Let $N(i) = \{j, ||\mathbf{b}_i - \mathbf{b}_j|| \leq r_i + r_j\}$ be the list of view-centered models j which may have intersection with the i^{th} view-centered model. Then the uncertainty-based filter is accelerated thanks to $N(i)$: triangle t is rejected if $U_i(\mathbf{v})/\min_{j \in N(i)} U_j(\mathbf{v}) \geq u_0$ for all vertices $\mathbf{v}$ of t. Last, the redundancy-based filter is applied. Its complexity due to uncertainty sort is $O(p\log(p))$, where p is the number of triangles. Its complexity due to redundancy triangle tests is $O(p^2)$, but this is accelerated using test eliminations and hierarchical bounding boxes.

3 EXPERIMENTS

The image sequence is taken in the university campus on august 15-16th afternoons without people. There are several trajectory loops, variable ground surface (road, foot path, unmown grass), buildings, corridor and vegetation (bushes, trees). This scene accumulates several difficulties: not 100% rigid scene (due to breath of wind), illuminations changes between day 1-2 subsequences (Fig 2), low-textured areas, camera gain changes (uncorrected), aperture problem and non-uniform sky at building-sky edges. The sequence has 2260 3264×2448 JPEG images, which are reduced by 2 in both dimensions to accelerate all calculations.

The perspective camera points toward the sky, it is hand-held and mounted on a monopod. The mirror (www.0 $-$360.com, 2010) provides large view field: 360 degrees in the horizontal plane, about 52 degrees above and 62 degrees below. The view field is projected between concentric circles of radii 572 and 103 pixels. We use a core 2 duo 2.5Ghz laptop with 4Go 667MHz DDR2.

First, the geometry is estimated thanks to the methods in Sections 2.1, 2.2 and 2.3. The user provides the list of image pairs $\{i, j\}$ such that drift between i and j should be removed (drift detection method is not integrated in the current version of the system). Once the geometry of days 1 and 2 sub-sequences are estimated using the initial calibration, points are matched between images i and j using correlation (Fig. 2) and CBA is applied to

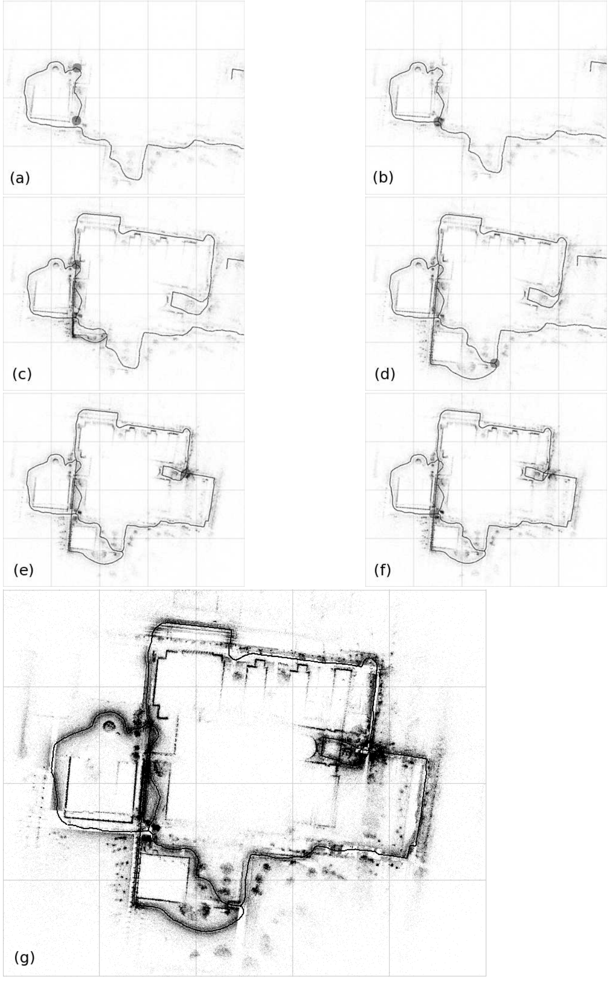

Figure 1: Geometry estimation steps: (a) day 1 sequence, (b) remove drift, (c) merge day 1-2 sequences, (d-f) remove drifts, (g) use all features. All results are registered in rectangle $[0,1] \times [0,0.8]$ by enforcing constant coordinates on the two poses surrounded by gray disks in (a). Gray disks in (b,d,e,f) show poses where drift is corrected. Day 1-2 sequences are merged on gray disk in (c).

remove drifts using $k = 1$. Cases (b,d,e,f) of Fig. 1 are trajectory loops with (424,451,1434,216) images and are obtained by (16,62,39,9) CBA iterations in (190,2400,1460,370) seconds, respectively. We think that a large part of the drift in case (d) is due to the single view point approximation, which is inaccurate in the outdoor corridor (top right corner of Fig. 4) with small scene depth. A last BA is applied to refine the geometry (3D and intrinsic parameters) and to increase the list of reconstructed points. The final geometry (Fig. 1.g) has 699410 points reconstructed from 3.9M Harris features; the means of track lengths and 3D points visible in one view are 5.5 and 1721, respectively.

Then, 2256 view-centered models are reconstructed thanks to the methods in Section 2.4 and 2.5 using $k = 1$. This is the most time consuming part of the method since one view-centered model is computed in about 3 min 30s. The first step of view-centered model computation is the over-segmentation mesh in the reference image. It samples the view field such that the super-pixels at the neighborhood of horizontal plane projection are initialized by squares of size 8×8 pixels in the images. The mean of number of 3D triangles is 17547. Fig. 3 shows super-pixels of a reference image and the resulting view-centered model.

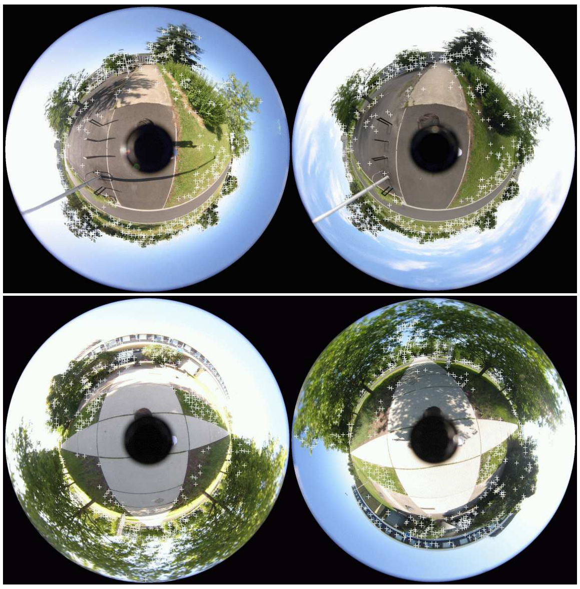

Figure 2: From top to bottom: 722 and 535 matches of $L_{i,j}$ used to remove drift in cases (d) and (e) of Fig. 1. Images of days 1 and 2 are on the left and right, respectively.

Last, the methods in Section 2.6 are applied to filter the 39.6M triangles stored in hard disk. A first filtering is done using reliability ($\alpha_0 = 5$ degrees), prior knowledge and uncertainty filters ($u_0 = 1.1$): we obtain 6.5M triangles in 40 min and store them in RAM. Redundancy removal is the last filtering and selects 4.5M triangles in 44 min. Texture packing and VRML file saving take 9 min. Fig. 4 shows views of the final model. We note that the scene is curved as if it lie on a sphere surface whose diameter has several kilometers: a vertical component of drift is left.

An other experiment is the quantitative evaluation of scene accuracy (discrepancy between scene reconstruction and ground truth) for a view-centered model using $k = 1$. A representative range of baselines is obtained with the following ground truth: the $[0,5]^3$ cube and camera locations defined by $\mathbf{c}_i = \begin{pmatrix}1 & 1+i/5 & 1\end{pmatrix}^T, i \in \{0,1,2\}$ (numbers in meters). First, synthetic images are generated using ray-tracing and the knowledge of mirror/perspective camera/textured cube. Second, methods in Sections 2.1, 2.2, 2.4 and 2.5 are applied. Third, a camera-based registration is applied to put the scene estimation in the coordinate frame of ground truth. Last, the scene accuracy $a_{0.9}$ is estimated using the distance e between vertex $\mathbf{v}$ of the model and the ground truth surface: inequality $|e(\mathbf{v})| \leq a_{0.9}||\mathbf{v} - \mathbf{c}_1||$ is true for 90% of vertices. We obtain $a_{0.9} = 0.015$.

4 CONCLUSION

We present an environment reconstruction system from images acquired by a $1000 camera. Several items are described: camera model, structure-from-motion, drift removal, view field sampling by super-pixels, view-centered model and triangle filtering. Unlike previous work, image meshes define both super-pixels (convex polygons) and triangles of 3D models. The current system is fully automatic up to the loop detection step (that previous methods could solve). Last it is experimented on a challenging sequence.

Future work includes loop detection integration, better use of visibility and prior knowledge for scene reconstruction, joining

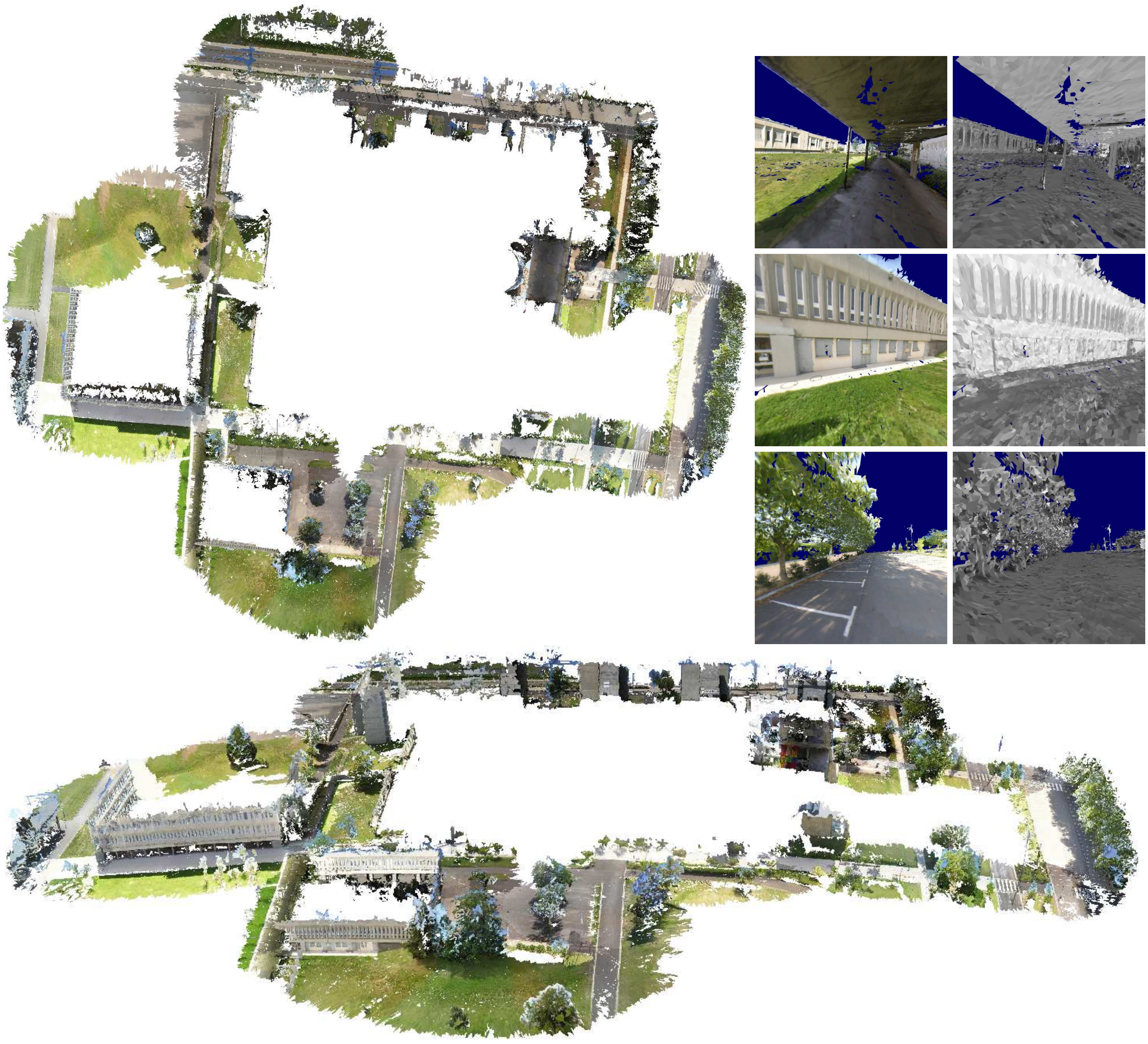

Figure 4: Top view (top left), local views (top right) and oblique view (bottom) of the final 3D model of the campus. The top view can be matched with Fig. 1.g. The transformation between top and oblique views is a rotation around horizontal axis.

meshes of view-centered models to form a continuous surface, and accelerations using GPU.

REFERENCES

Anan, C. S. and Hartley, R., 2005. Visual localization and loop-back detection with a high resolution omnidirectional camera. In: OMNIVIS Workshop.

Chai, B., Sethuraman, S., Sawhney, H. and Hatrack, P., 2004. Depth map compression for real-time view-based rendering. Pattern recognition letters 25(7), pp. 755–766.

Cornelis, K., Verbiest, F. and Gool, L. V., 2004. Drift removal for sequential structure from motion algorithms. PAMI 26(10), pp. 1249–1259.

Doubek, P. and Svoboda, T., 2002. Reliable 3d reconstruction from a few catadioptric images. In: OMNIVIS Workshop.

Havlena, M., Torri, A., Knopp, J. and Pajdla, T., 2009. Randomized structure from motion based on atomic 3d models from camera triplets. In: CVPR'09.

Lhuillier, M., 2008a. Automatic scene structure and camera motion using a catadioptric system. CVIU 109(2), pp. 186–203.

Lhuillier, M., 2008b. Toward automatic 3d modeling of scenes using a generic camera model. In: CVPR'08.

Lhuillier, M. and Quan, L., 2002. Match propagation for image-based modeling and rendering. PAMI 24(8), pp. 1140–1146.

Micusik, B. and Kosecka, J., 2009. Piecewise planar city 3d modeling from street view panoramic sequence. In: CVPR'09.

Pollefeys, M., Nister, D., Frahm, J., Akbarzadeh, A., Mordohai, P., Clipp, B., Engels, C., Gallup, D., Kim, S., Merell, P., Salmi, C., Sinha, S., Talton, B., Wang, L., Yang, Q., Stewenius, H., Yang, R., Welch, G. and Towles, H., 2008. Detailed real-time urban 3d reconstruction from video. IJCV 78(2), pp. 143–167.

Schindler, K. and Bischof, H., 2003. On robust regression in photogrammetric point clouds. In: DAGM'03.

Triggs, B., McLauchlan, P., Hartley, R. and Fitzgibbon, A., 2000. Bundle adjustment – a modern synthesis. In: Vision Algorithms Workshop.

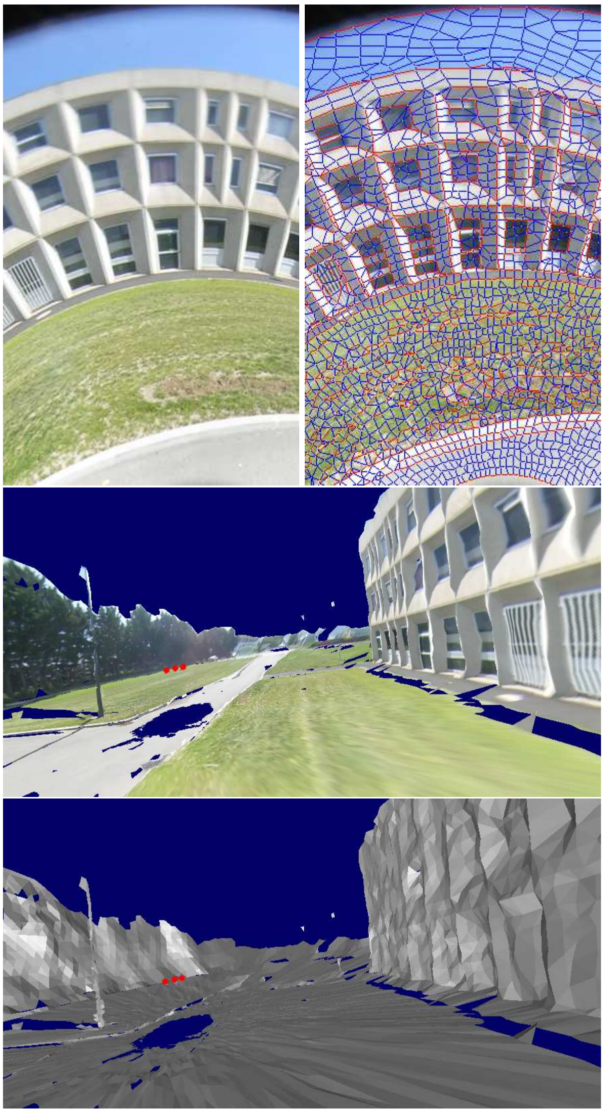

Figure 3: From top to bottom: part of reference image, over-segmentation using polygons (contours in red), view-centered model (texture and normals) reconstructed from 3 poses (in red).

www.0 −360.com, 2010.

www.ptgrey.com, 2010.

Zitnick, C. and Kang, S., 2007. Stereo for image-based rendering using image over-segmentation. IJCV 75(1), pp. 49–65.

APPENDIX A: POINT AND PLANE THRESHOLDING

Let $\mathbf{p}$ be a 3D point. The covariance matrix $\mathtt{C}_{\mathbf{p}}$ of $\mathbf{p}$ is provided by ray intersection from $\mathbf{p}$ projections in images $\mathcal{N}(i) = \{i-k \cdots i \cdots i+k\}$ using Levenberg-Marquardt. In this paper, ray intersection and covariance $\mathtt{C}_{\mathbf{p}}$ result from the angle error in (Lhuillier, 2008b). The Mahalanobis distance $D_{\mathbf{p}}$ between points $\mathbf{p}$ and $\mathbf{p}'$ is $D_{\mathbf{p}}(\mathbf{p}') = \sqrt{(\mathbf{p}-\mathbf{p}')^T \mathtt{C}_{\mathbf{p}}^{-1}(\mathbf{p}-\mathbf{p}')}$.

We define points $\mathbf{p} = \mathbf{c}_i + z\mathbf{u}$ and $\mathbf{p}' = \mathbf{c}_i + z'\mathbf{u}$ using camera location $\mathbf{c}_i$, ray direction $\mathbf{u}$ and depths z, z'. If z is large enough, $\mathbf{u}$ is a good approximation of the main axis of $\mathtt{C}_{\mathbf{p}}$: we have $\mathtt{C}_{\mathbf{p}} \approx \sigma_{\mathbf{p}}^2 \mathbf{u}\mathbf{u}^T$ and $\mathbf{u}^T \mathtt{C}_{\mathbf{p}}^{-1} \mathbf{u} \approx \sigma_{\mathbf{p}}^{-2}$ where $\sigma_{\mathbf{p}}^2$ is the largest singular value of $\mathtt{C}_{\mathbf{p}}$. In this context, we obtain $D_{\mathbf{p}}(\mathbf{p}') \approx \frac{|z-z'|}{\sigma_{\mathbf{p}}}$.

If $\mathbf{x}$ has the Gaussian distribution with mean $\mathbf{p}$ and covariance $\mathtt{C}_{\mathbf{p}}$, $D_{\mathbf{p}}^2(\mathbf{x})$ has the $\mathcal{X}^2$ distribution with 3 d.o.f. We decide that points $\mathbf{p}$ and $\mathbf{p}'$ are the same (up to error) if both $D_{\mathbf{p}}^2(\mathbf{p}')$ and $D_{\mathbf{p}'}^2(\mathbf{p})$ are less than the 90% quantile of this distribution: we decide that $\mathbf{p}$ and $\mathbf{p}'$ are the same point if $D(\mathbf{p},\mathbf{p}') \leq 2.5$ where $D(\mathbf{p},\mathbf{p}') = \max\{D_{\mathbf{p}}(\mathbf{p}'), D_{\mathbf{p}'}(\mathbf{p})\} \approx \frac{|z-z'|}{\min\{\sigma_{\mathbf{p}},\sigma_{\mathbf{p}'}\}}$.

Let π be the plane $\mathbf{n}^T\mathbf{x}+d = 0$. The point-to-plane Mahalanobis distance is $D_{\mathbf{p}}^2(\pi) = \min_{\mathbf{x}\in\pi} D_{\mathbf{p}}^2(\mathbf{x}) = \frac{(\mathbf{n}^T\mathbf{p}+d)^2}{\mathbf{n}^T\mathtt{C}_{\mathbf{p}}\mathbf{n}}$ (Schindler and Bischof, 2003). Thus $\mathtt{C}_{\mathbf{p}} \approx \sigma_{\mathbf{p}}^2\mathbf{u}\mathbf{u}^T$ and $\mathbf{p}' \in \pi$ imply $D_{\mathbf{p}}^2(\pi) \approx \frac{(\mathbf{n}^T\mathbf{p}'+d+\mathbf{n}^T(\mathbf{p}-\mathbf{p}'))^2}{\sigma_p^2(\mathbf{n}^T\mathbf{u})^2} = \frac{(z-z')^2}{\sigma_{\mathbf{p}}^2} \approx D_{\mathbf{p}}^2(\mathbf{p}')$.

Last, we obtain the point-to-plane thresholding and distance used in Section 2.5. We decide that $\mathbf{p}$ is in plane π if $D(\mathbf{p},\mathbf{p}') \leq 2.5$ where $\mathbf{p}' \in \pi$. The robust distance between $\mathbf{p}$ and π is $\min\{D(\mathbf{p},\mathbf{p}'), 2.5\} \approx \min\{\frac{|z-z'|}{\min\{\sigma_{\mathbf{p}},\sigma_{\mathbf{p}'}\}}, 2.5\}$, $z' = -\frac{\mathbf{n}^T\mathbf{c}_i+d}{\mathbf{n}^T\mathbf{u}}$.

APPENDIX B: CONSTRAINED BUNDLE ADJUSTMENT

In Section 2.3, we would like to apply CBA (constrained bundle adjustment) summarized in (Triggs et al., 2000) to remove the drift. This method minimizes the re-projection error function $\mathbf{x} \mapsto f(\mathbf{x})$ subject to the drift removal constraint $c(\mathbf{x}) = 0$, where $\mathbf{x}$ concatenates poses and 3D points. Here we have $c(\mathbf{x}) = \mathbf{x}_1 - \mathbf{x}_1^g$ where $\mathbf{x}_1$ and $\mathbf{x}_1^g$ concatenate 3D locations of images $\mathcal{N}(i)$ and their duplicates of images $\mathcal{N}(n+k)$, respectively. All 3D parameters of sequence $\{0 \cdots n+2k\}$ are in $\mathbf{x}$ except the 3D locations of $\mathcal{N}(j)$ and $\mathcal{N}(n+k)$. During CBA, $\mathbf{x}_1^g$ is fixed and $\mathbf{x}_1$ evolves towards $\mathbf{x}_1^g$.

However, there is a difficulty with this scheme. CBA iteration (Triggs et al., 2000) improves $\mathbf{x}$ by adding step $\mathbf{\Delta}$ which minimizes quadratic Taylor expansion of f subject to $\mathbf{0} \approx c(\mathbf{x}+\mathbf{\Delta})$ and linear Taylor expansion $c(\mathbf{x}+\mathbf{\Delta}) \approx c(\mathbf{x}) + \mathtt{C}\mathbf{\Delta}$. We use notations $\mathbf{x}^T = \begin{pmatrix}\mathbf{x}_1^T & \mathbf{x}_2^T\end{pmatrix}$, $\mathbf{\Delta}^T = \begin{pmatrix}\mathbf{\Delta}_1^T & \mathbf{\Delta}_2^T\end{pmatrix}$, $\mathtt{C} = \begin{pmatrix}\mathtt{C}_1 & \mathtt{C}_2\end{pmatrix}$ and obtain $\mathtt{C}_1 = \mathtt{I}, \mathtt{C}_2 = \mathtt{0}$. Thus, we have $\mathbf{\Delta}_1 = -c(\mathbf{x})$ at the first CBA iteration. On the one hand, $\mathbf{\Delta}_1 = -c(\mathbf{x})$ is the drift and may be very large. On the other hand, $\mathbf{\Delta}$ should be small enough for quadratic Taylor approximation of f.

The “reduced problem” in (Triggs et al., 2000) is used: BA iteration minimizes the quadratic Taylor expansion of $\mathbf{\Delta}_2 \mapsto g(\mathbf{\Delta}_2)$ where $g(\mathbf{\Delta}_2) = f(\Delta(\mathbf{\Delta}_2))$ and $\Delta(\mathbf{\Delta}_2)^T = \begin{pmatrix}-c(\mathbf{x})^T & \mathbf{\Delta}_2^T\end{pmatrix}$. Step $\mathbf{\Delta}_2$ meets $H_2(\lambda)\mathbf{\Delta}_2 = -\mathbf{g}_2$, where $(\lambda, \mathbf{g}_2, H_2(\lambda))$ are damping parameter, gradient and damped hessian of g. Update $\mathbf{x} \leftarrow \mathbf{x} + \Delta(\mathbf{\Delta}_2)$ holds if $g(\mathbf{\Delta}_2) < \min\{1.1f_0, g(\mathbf{0})\}$, where f_0 is the value of $f(\mathbf{x})$ before CBA. It can be shown that this inequality is true if $c(\mathbf{x})$ is small enough and λ is large enough.

Here we reset c by c_n at the n^{th} iteration of CBA to have a small enough $c(\mathbf{x})$. Let $\mathbf{x}_1^0$ be the value of $\mathbf{x}_1$ before CBA. We use $c_n(\mathbf{x}) = \mathbf{x}_1 - ((1-\gamma_n)\mathbf{x}_1^0 + \gamma_n\mathbf{x}_1^g)$, where γ_n increases progressively from 0 (no constraint at CBA start) to 1 (full constraint at CBA end). One CBA iteration is summarized as follows. First, estimate $\mathbf{\Delta}_2(\gamma_n)$ for the current value of $(\lambda, \mathbf{x})$ (a single linear system $H_2(\lambda)\mathtt{X} = \mathtt{Y}$ is solved for all $\gamma_n \in [0,1]$). Second, try to increase γ_n such that $g(\mathbf{\Delta}_2(\gamma_n)) < \min\{1.1f_0, g(\mathbf{0})\}$. If the iteration succeeds, apply $\mathbf{x} \leftarrow \mathbf{x} + \Delta(\mathbf{\Delta}_2)$. Furthermore, apply $\lambda \leftarrow \lambda/10$ if $\gamma_n = \gamma_{n-1}$. If the iteration fails, apply $\lambda \leftarrow 100\lambda$. If $\gamma_n > \gamma_{n-1}$ or $\gamma_n = 1$, choose $\gamma_{n+1} = \gamma_n$ at the $(n+1)^{th}$ iteration to obtain $\Delta(\mathbf{\Delta}_2)^T = \begin{pmatrix}\mathbf{0}^T & \mathbf{\Delta}_2^T\end{pmatrix}$ and to decrease f as soon (or much) as possible.